

10-06-00

A

**UTILITY PATENT APPLICATION TRANSMITTAL**  
**(Large Entity)***(Only for new nonprovisional applications under 37 CFR 1.53(b))*Docket No.  
13768.169Total Pages in this Submission  
44**TO THE ASSISTANT COMMISSIONER FOR PATENTS**Box Patent Application  
Washington, D.C. 20231

Transmitted herewith for filing under 35 U.S.C. 111(a) and 37 C.F.R. 1.53(b) is a new utility patent application for an invention entitled:

**TRANSPARENTLY REDIRECTING CLIENT REQUESTS FOR CONTENT**

and invented by:

Joel Soderberg  
Brian Deen  
Alex HopmannIf a **CONTINUATION APPLICATION**, check appropriate box and supply the requisite information:☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: \_\_\_\_\_

Which is a:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: \_\_\_\_\_

Which is a:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: \_\_\_\_\_

Enclosed are:

**Application Elements**

1. ☒ Filing fee as calculated and transmitted as described below
2. ☒ Specification having 34 pages and including the following:
  - a. ☒ Descriptive Title of the Invention
  - b. ☒ Cross References to Related Applications *(if applicable)*
  - c. ☒ Statement Regarding Federally-sponsored Research/Development *(if applicable)*
  - d. ☒ Reference to Microfiche Appendix *(if applicable)*
  - e. ☒ Background of the Invention
  - f. ☒ Brief Summary of the Invention
  - g. ☒ Brief Description of the Drawings *(if drawings filed)*
  - h. ☒ Detailed Description
  - i. ☒ Claim(s) as Classified Below
  - j. ☒ Abstract of the Disclosure

**UTILITY PATENT APPLICATION TRANSMITTAL**  
**(Large Entity)**

*(Only for new nonprovisional applications under 37 CFR 1.53(b))*

Docket No.  
13768.169

Total Pages in this Submission  
44

**Application Elements (Continued)**

3. ☒ Drawing(s) *(when necessary as prescribed by 35 USC 113)*
- a. ☒ Formal                      Number of Sheets 5
- b. ☐ Informal                      Number of Sheets \_\_\_\_\_
4. ☐ Oath or Declaration
- a. ☐ Newly executed *(original or copy)*                      ☐ Unexecuted
- b. ☐ Copy from a prior application (37 CFR 1.63(d)) *(for continuation/divisional application only)*
- c. ☐ With Power of Attorney                      ☐ Without Power of Attorney
- d. ☐ DELETION OF INVENTOR(S)  
Signed statement attached deleting inventor(s) named in the prior application,  
see 37 C.F.R. 1.63(d)(2) and 1.33(b).
5. ☐ Incorporation By Reference *(usable if Box 4b is checked)*  
The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under  
Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby  
incorporated by reference therein.
6. ☐ Computer Program in Microfiche *(Appendix)*
7. ☐ Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all must be included)*
- a. ☐ Paper Copy
- b. ☐ Computer Readable Copy *(identical to computer copy)*
- c. ☐ Statement Verifying Identical Paper and Computer Readable Copy

**Accompanying Application Parts**

8. ☐ Assignment Papers *(cover sheet & document(s))*
9. ☐ 37 CFR 3.73(B) Statement *(when there is an assignee)*
10. ☐ English Translation Document *(if applicable)*
11. ☐ Information Disclosure Statement/PTO-1449                      ☐ Copies of IDS Citations
12. ☐ Preliminary Amendment
13. ☒ Acknowledgment postcard
14. ☒ Certificate of Mailing
- ☐ First Class    ☒ Express Mail *(Specify Label No.):* EL695574992US

**UTILITY PATENT APPLICATION TRANSMITTAL**  
**(Large Entity)**

*(Only for new nonprovisional applications under 37 CFR 1.53(b))*

Docket No.  
13768.169

Total Pages in this Submission  
44

**Accompanying Application Parts (Continued)**

15. ☐ Certified Copy of Priority Document(s) *(if foreign priority is claimed)*

16. ☒ Additional Enclosures *(please identify below):*

Form PTO-2038 submitting payment in the amount of \$1,184  
Attachement for correspondence

**Request That Application Not Be Published Pursuant To 35 U.S.C. 122(b)(2)**

17. ☐ Pursuant to 35 U.S.C. 122(b)(2), Applicant hereby requests that this patent application not be published pursuant to 35 U.S.C. 122(b)(1). Applicant hereby certifies that the invention disclosed in this application has not and will not be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication of applications 18 months after filing of the application.

**Warning**

***An applicant who makes a request not to publish, but who subsequently files in a foreign country or under a multilateral international agreement specified in 35 U.S.C. 122(b)(2)(B)(i), must notify the Director of such filing not later than 45 days after the date of the filing of such foreign or international application. A failure of the applicant to provide such notice within the prescribed period shall result in the application being regarded as abandoned, unless it is shown to the satisfaction of the Director that the delay in submitting the notice was unintentional.***

**UTILITY PATENT APPLICATION TRANSMITTAL**  
**(Large Entity)**

*(Only for new nonprovisional applications under 37 CFR 1.53(b))*

Docket No.  
13768.169

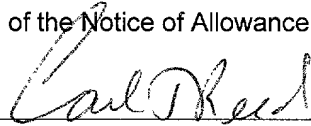
Total Pages in this Submission  
44

**Fee Calculation and Transmittal**

**CLAIMS AS FILED**

For	#Filed	#Allowed	#Extra	Rate	Fee
Total Claims	33	- 20 =	13	x \$18.00	\$234.00
Indep. Claims	6	- 3 =	3	x \$80.00	\$240.00
Multiple Dependent Claims (check if applicable) <input type="checkbox"/>					\$0.00
BASIC FEE					\$710.00
OTHER FEE (specify purpose)					\$0.00
TOTAL FILING FEE					\$1,184.00

- ☒ A check in the amount of \_\_\_\_\_ to cover the filing fee is enclosed.
- ☒ The Commissioner is hereby authorized to charge and credit Deposit Account No. 23-3178 as described below. A duplicate copy of this sheet is enclosed.
- ☐ Charge the amount of \_\_\_\_\_ as filing fee.
- ☒ Credit any overpayment.
- ☒ Charge any additional filing fees required under 37 C.F.R. 1.16 and 1.17.
- ☐ Charge the issue fee set in 37 C.F.R. 1.18 at the mailing of the Notice of Allowance, pursuant to 37 C.F.R. 1.311(b).

  
Signature

Carl T. Reed  
Registration No. 45,454

Dated: October 4, 2000



**022913**

CC:

PATENT TRADEMARK OFFICE

# UNITED STATES PATENT APPLICATION

of

**Joel Soderberg**

## Brian Deen

and

**Alex Hopmann**

for

## TRANSPARENTLY REDIRECTING CLIENT REQUESTS FOR CONTENT

## **BACKGROUND OF THE INVENTION**

### **1. Co-pending Applications Containing Related Material**

The following co-pending applications, which are filed the same day as this application, are hereby incorporated by reference: U.S. Application No. \_\_/\_\_\_\_\_, entitled “Selecting a Server to Service Client Requests,” and U.S. Application No. \_\_/\_\_\_\_\_, entitled “Routing Client Requests to Back-End Servers.”

### **2. The Field of the Invention**

The present invention relates to redirecting requests for content that is stored on a server. More specifically, the present invention relates to methods and computer program products for using a front-end server in transparently redirecting a request for content such that a client system is unaware of the redirection.

### **3. The Prior State of the Art**

At times, a server may need to redirect requests for content. Redirection may occur because (i) the content has moved, either temporarily or permanently, to another server, (ii) multiple representations of the content are available and the client should choose from them, (iii) the request should be made through a proxy, and etc. In the prior art, servers send redirect responses to the client system and the client system is responsible for reissuing the request for content to the server identified in the redirect response. However, sending redirect responses to a client may not provide any benefit in some systems.

For example, some systems include a proxy server that provides a single point of access to content stored on back end servers. The back end server is a server in the ordinary sense, storing content that clients may access. The proxy or front end server provides an

1 access point for the content that is stored on one or more back end servers. From the client's  
2 perspective, it appears as if the front end server is the source for all of the content stored at  
3 the back end servers, However, clients are not necessarily precluded from accessing back  
4 end servers directly.

5 Making requests for content through front-end servers provides several benefits.  
6 From time to time, content stored on one back-end server may be moved to another. If a  
7 client has been accessing the moved content directly, the client must alter future requests to  
8 reflect the new location of the content. Although redirect responses from the old back-end  
9 server may help inform clients that the content has moved, redirection requires additional  
10 processing by the client and leads to slower response times for the requested content. Also,  
11 redirect responses from back-end servers may be provided on a temporary basis. Requests  
12 occurring after a certain time period may indicate simply that the requested content is not  
13 available.

14 Front-end servers also facilitate local caching at the client system. Frequently,  
15 content, such as email content, is cached at the client system to improve performance. Only  
16 changes made to the email content is exchanged between the client and back-end server. If  
17 email content is moved from one back-end server to another, the local caching is invalidated  
18 and the full email content must be exchanged. By using a front-end server, all content  
19 appears to come from a single source, regardless of the particular back-end server that  
20 actually stores the content. Any movement of content from one back-end server to another  
21 is transparent to the client. As a result, local caching remains valid even after content has  
22 been moved.

23 However, redirection, as taught in the prior art, introduces problems when practiced  
24 in a front-end server system. First, the front-end server is supposed to make the storage of

1 content at back-end servers transparent to the client. In receiving a redirect response  
2 because content has moved from one back-end server to another, the client is made aware of  
3 back-end server storage details and transparent access to the content is eliminated.  
4 Furthermore, because the client made the request to the front-end server, a redirect response  
5 from a back-end server that is unknown to the client is likely to cause substantial confusion.  
6 Even more troublesome, certain back-end servers may require that all access occur through a  
7 front-end server. In these circumstances, the client would be redirected to a back-end server  
8 that the client is unable to access. Therefore, the prior art lacks methods and computer  
9 program products for using a front-end server in transparently redirecting a request for  
10 content such that a client system is unaware of the redirection.



## SUMMARY OF THE INVENTION

These and other problems with the prior art are overcome by the present invention, which is directed toward transparently redirecting a request for content such that a client system is unaware of the redirection. A front-end server receives requests for content stored on one or more back-end servers. By making the requests through the front-end server, the client does not know and does not care which back-end server stores the content being requested. As far as the client is concerned, the front-end server is the source of the requested content.

Content may be stored on one or more back-end servers. Private content, such as email, calendars, tasks, etc., typically is stored at only one back-end server, whereas public content, like applications, discussion groups, etc., typically is stored on multiple servers. At times, a front-end server may request data from a back-end server that the back-end server does not store. For example, if private content is moved, the back-end server generates a redirect response that identifies the back-end server storing the moved content. Rather than pass the redirect response back to the client as is taught in the prior art, the front-end server reissues the content request to the back-end server identified in the redirect response. Upon receiving the requested content from the redirect back-end server, the front-end sends the content to the client system.

Public content is likely to be stored on multiple back-end servers. However, the prior art allows for only a single server to be identified in a redirect response. When the front-end server receives a request for content from a client, the request is modified to indicate that the front-end is making the request to the back-end server. The back-end server receiving the request for content examines the request to determine if the request includes the front-end indicator. If present, the back-end server identifies a list of back-end servers

1 that store the requested content. By receiving a list of back-end servers, the front-end server  
2 is able to load balance requests for content among all the back-end servers that can satisfy  
3 the request. If the front-end indicator is not present, the back-end server identifies a single  
4 back-end server in the redirect response. In identifying only a single back-end server, the  
5 redirect response will be compatible with prior art clients making direct requests to back-end  
6 servers.

7 Because the front-end server processes redirections, rather than passing them to the  
8 client as taught in the prior art, the front-end server preserves the benefits of having  
9 transparent back-end server storage. Specifically, content can move from one back-end  
10 server to another without impacting the client.

11 Additional features and advantages of the invention will be set forth in the  
12 description which follows, and in part will be obvious from the description, or may be  
13 learned by the practice of the invention. The features and advantages of the invention may  
14 be realized and obtained by means of the instruments and combinations particularly pointed  
15 out in the appended claims. These and other features of the present invention will become  
16 more fully apparent from the following description and appended claims, or may be learned  
17 by the practice of the invention as set forth hereinafter.

**BRIEF DESCRIPTION OF THE DRAWINGS**

In order to describe the manner in which the above-recited and other advantages and features of the invention can be obtained, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

Figure 1 illustrates an exemplary system that provides a suitable operating environment for the present invention;

Figure 2 shows a client system accessing email content that has moved from one back-end server to another;

Figure 3 show a client system accessing content that is stored at various back-end servers through a front-end server; and

Figures 4A and 4B are flowcharts illustrating exemplary methods for redirecting requests for content stored at a back-end server according to the present invention.

## DETAILED DESCRIPTION OF THE INVENTION

Client requests for data may be redirected from one server to another. Some redirections occur because data has been either temporarily or permanently moved to a new server. In the prior art, redirections are sent to the client system that initiated the request and the client reissues the request to the server indicated in the redirection. Where a proxy requests data for a client, the proxy returns redirection responses to the client for processing. However, redirection according to the prior art proves to be inadequate for certain systems. For example, U.S. Application No. \_\_/\_\_\_\_\_, filed on the same day herewith, entitled "Routing Client Requests to Back-End Servers," which is hereby incorporated by reference, describes the use of a front-end server to access content stored on one or more back-end servers. The specific back-end server storing requested content is unknown and transparent to the client.

One benefit of accessing content through a front-end server as described in the afore-referenced application is that the client system is insulated from having to identify the specific server storing the data being requested. The client requests data from the front-end server and the front-end server determines which back-end server should satisfy the request. Use of the front-end server also insures that local caching on the client remains valid when content moves from one back-end server to another because it appears to the client that the front-end server is the source of content.

With all requests for content being directed to the front-end server, redirection presents a problem because it identifies specific back-end servers. For example, a redirect response may not provide any meaningful information. In some systems, clients may be allowed access to back-end servers only through a front-end server. This makes the redirection information of no use to the client because reissuing the request directly to the

1 identified back-end server will result in the request being denied. Even if the back-end  
2 server is directly accessible to the client, a direct request to the back-end server prevents the  
3 client from receiving any of the benefits offered by the front-end server arrangement.

4 In a Microsoft Exchange ® system, redirections may occur for a variety of reasons.  
5 For example, Exchange public folder trees may be replicated across multiple back-end  
6 servers to provide greater accessibility. However, the extent of content replication (in  
7 contrast to folder tree replication) is controlled on a folder-by-folder basis. As a result, a  
8 given back-end server may include the folder hierarchy, but not the folder contents. A  
9 folder is termed “ghosted” when the folder is present in the public folder tree of a back-end  
10 server, but the contents are not replicated on the back-end server.

11 The front-end server may direct requests for content based on the public folder tree.  
12 If a request is for the contents of a folder that is ghosted, the back-end server issues a  
13 redirect response that identifies other back-end servers where the content is replicated.  
14 According to the present invention, the front-end server receives the redirect response from  
15 the back-end server and reissues the redirected request rather than passing the redirect  
16 response back to the client. When the front-end server receives the requested content from a  
17 back-end server identified in the redirect response, the content is sent to the client. From the  
18 client’s perspective, the content’s source is the front-end server. The client has no indication  
19 that the redirection occurred.

20 Because replicated data may be stored on more than one server, the back-end server  
21 ghosting a folder is capable of identifying a list of back-end servers in a redirect response.  
22 However, prior art clients are unable to receive a list of servers in a redirect response. As a  
23 result, front-end servers add a front-end indicator to any requests they direct to back-end  
24 servers. If a back-end server receives a request with the front-end indicator, the back-end



Figure 1 and the following discussion are intended to provide a brief, general description of a suitable computing environment in which the invention may be

With reference to Figure 1, an exemplary system for implementing the invention includes a general purpose computing device in the form of a conventional computer 20, including a processing unit 21, a system memory 22, and a system bus 23 that couples various system components including the system memory 22 to the processing unit 21. The system bus 23 may be any of several types of bus structures including a memory bus or



1 memory controller, a peripheral bus, and a local bus using any of a variety of bus  
2 architectures. The system memory includes read only memory (ROM) 24 and random  
3 access memory (RAM) 25. A basic input/output system (BIOS) 26, containing the basic  
4 routines that help transfer information between elements within the computer 20, such as  
5 during start-up, may be stored in ROM 24.

6 The computer 20 may also include a magnetic hard disk drive 27 for reading from  
7 and writing to a magnetic hard disk 39, a magnetic disk drive 28 for reading from or writing  
8 to a removable magnetic disk 29, and an optical disk drive 30 for reading from or writing to  
9 removable optical disk 31 such as a CD-ROM or other optical media. The magnetic hard  
10 disk drive 27, magnetic disk drive 28, and optical disk drive 30 are connected to the system  
11 bus 23 by a hard disk drive interface 32, a magnetic disk drive-interface 33, and an optical  
12 drive interface 34, respectively. The drives and their associated computer-readable media  
13 provide nonvolatile storage of computer-executable instructions, data structures, program  
14 modules and other data for the computer 20. Although the exemplary environment  
15 described herein employs a magnetic hard disk 39, a removable magnetic disk 29 and a  
16 removable optical disk 31, other types of computer readable media for storing data can be  
17 used, including magnetic cassettes, flash memory cards, digital video disks, Bernoulli  
18 cartridges, RAMs, ROMs, and the like.

19 Program code means comprising one or more program modules may be stored on the  
20 hard disk 39, magnetic disk 29, optical disk 31, ROM 24 or RAM 25, including an operating  
21 system 35, one or more application programs 36, other program modules 37, and program  
22 data 38. A user may enter commands and information into the computer 20 through  
23 keyboard 40, pointing device 42, or other input devices (not shown), such as a microphone,  
24 joy stick, game pad, satellite dish, scanner, or the like. These and other input devices are

1 often connected to the processing unit 21 through a serial port interface 46 coupled to  
2 system bus 23. Alternatively, the input devices may be connected by other interfaces, such  
3 as a parallel port, a game port or a universal serial bus (USB). A monitor 47 or another  
4 display device is also connected to system bus 23 via an interface, such as video adapter 48.  
5 In addition to the monitor, personal computers typically include other peripheral output  
6 devices (not shown), such as speakers and printers.

7 The computer 20 may operate in a networked environment using logical connections  
8 to one or more remote computers, such as remote computers 49a and 49b. Remote  
9 computers 49a and 49b may each be another personal computer, a server, a router, a network  
10 PC, a peer device or other common network node, and typically include many or all of the  
11 elements described above relative to the computer 20, although only memory storage  
12 devices 50a and 50b and their associated application programs 36a and 36b have been  
13 illustrated in Figure 1. The logical connections depicted in Figure 1 include a local area  
14 network (LAN) 51 and a wide area network (WAN) 52 that are presented here by way of  
15 example and not limitation. Such networking environments are commonplace in office-  
16 wide or enterprise-wide computer networks, intranets and the Internet.

17 When used in a LAN networking environment, the computer 20 is connected to the  
18 local network 51 through a network interface or adapter 53. When used in a WAN  
19 networking environment, the computer 20 may include a modem 54, a wireless link, or other  
20 means for establishing communications over the wide area network 52, such as the Internet.  
21 The modem 54, which may be internal or external, is connected to the system bus 23 via the  
22 serial port interface 46. In a networked environment, program modules depicted relative to  
23 the computer 20, or portions thereof, may be stored in the remote memory storage device. It  
24

1 will be appreciated that the network connections shown are exemplary and other means of  
2 establishing communications over wide area network 52 may be used.

3 Figure 2 shows a client system accessing email content that has moved from one  
4 back-end server to another. Email server A 250 includes mailbox storage A 252 containing  
5 email client A new mailbox 254, whereas email server B 260 includes mailbox storage  
6 B 262 containing email client A old mailbox 264. Email server C 270 includes mailbox  
7 storage C 272. Email server A 250, email server B 260, and email server C 270 are  
8 examples of back-end servers. Email client A new mailbox 254 and email client A old  
9 mailbox 264 are examples of resources available through front-end server 240. Global  
10 catalog server 230 provides front-end server 240 with current information regarding the  
11 content stored at each of the available back-end servers. Additionally, the front-end server  
12 240 is stateless and does not require, for example, a hard drive for storing program data.  
13 U.S. Application No. \_\_/\_\_\_\_\_, filed on the same day herewith, and entitled "Selecting a  
14 Server to Service Client Requests," which is hereby incorporated by reference, provides  
15 greater detail with respect to how a particular back-end server may be identified and how a  
16 front-end server operates.

17 Email client A 210 uses front-end server 240 in accessing email content. Various  
18 protocols are available that allow front-end server 240 to route content requests on behalf of  
19 email client A 210. Internet message access protocol 4 ("IMAP4"), post office protocol  
20 version 3 ("POP3"), and hypertext transfer protocol ("HTTP") are all protocols that  
21 front-end server 240 can direct to a particular back-end server on behalf of email client  
22 A 210. The present invention, however, is not limited to any particular protocol. It is only  
23 necessary that the protocol allow for requests to be routed by a front-end server. For  
24

1 example, messaging application programming interface ("MAPI") is a protocol that  
2 currently will not allow front-end server 240 to route requests for email client A 210.

3 Comparing requests for content by email client A 210 with and without front-end  
4 server 240 may be helpful in understanding the present invention. For simplicity, assume  
5 that all requests use HTTP as their email protocol. Without front-end server 240, a request  
6 by email client A 210 for the contents of the email client A old mailbox 264 might be  
7 entered as "http://serverB/exchange/clientA" and a request by email client A 210 for the  
8 contents of the email client A new mailbox 254 might be entered as  
9 "http://serverA/exchange/clientA". Note that in each case, it was necessary to specify the  
10 server where the mailbox contents were stored. In contrast, if the front-end server 240 were  
11 named "mail" a request by email client A 210 for the contents of the email client A new  
12 mailbox 254 might be entered as "http://mail/exchange/clientA".

13 By using front-end server 240, email client A 210 does not know and does not care  
14 what back-end server stores the requested email content. When the email content is moved  
15 from email client A old mailbox 264 on server B 260 to email client A new mailbox 254 on  
16 server A 250, front-end server 240 will direct the request to the appropriate back-end server  
17 automatically. Nevertheless, there may be a short time delay between when email content is  
18 moved from server B 260 to server A 250 and when global catalog server 230 reflects the  
19 move. During this time delay, front-end server 240 continues to direct requests for the email  
20 content of email client A 210 to server B 260. Upon receiving a request, server B 260  
21 generates a redirect response that indicates that the email content for email client A 210 now  
22 is located at server A 250.

23 In contrast to the prior art, when front-end server 240 receives the redirect response,  
24 front-end server 240 reissues the request for email content to server A 250 rather than

1 passing the redirect response back to email client A 210. As shown in Figure 2, email client  
2 A 210 may only access email content through front-end server 240. A redirect response  
3 indicating that server A 250 stores the email content from email client A 210 would be of no  
4 benefit to email client A 210 because server A 250 cannot be reached directly. Furthermore,  
5 such a redirect response may confuse email client A 210 because no request for content was  
6 made to server B 260, and therefore a redirect response from server B 260 is unexpected.  
7 Front-end server 240 overcomes these problems with prior art redirection by processing the  
8 redirect response without any indication to email client A 210 that any redirection has  
9 occurred. U.S. Application No. \_\_/\_\_\_\_, filed on the same day herewith, entitled  
10 "Routing Client Requests to Back-End Servers," which is hereby incorporated by reference,  
11 provides further details on the operation of front-end server 240 in requesting email content  
12 for email client A 210. The present invention should not be interpreted as requiring that  
13 back-end server only be accessible through a front-end server.

14 The present invention is not limited to any particular protocol and it is not limited to  
15 any particular content. Figure 3 shows client system 310 using front-end server 340 to  
16 access content that is stored at various back-end servers. Back-end server 350 includes web  
17 store 352 containing App A 382 and Folder A 392 that stores Data A 394. Back-end server  
18 360 includes web store 362 containing App B 384 and Folder A 392 that stores Data A 394.  
19 Back-end server 370 includes web store 372 containing App A 382 and Folder A 392.  
20 However, Folder A 392 is dashed in web store 372 to indicate that the storage hierarchy is  
21 present at back-end server 370, but the data (e.g., Data A 394) is not.

22 App A 382, App B 384, and Folder A 392 are examples of public content stored at a  
23 back-end server. App A 382 is available in web stores 352 and 372, whereas App B 384 is  
24 only available in web store 362. Folder A 392 represents a storage hierarchy for organizing

1 public content. Microsoft Exchange ® products allow the determination of whether data  
2 corresponding to a folder will be stored on a particular server to be made on a  
3 folder-by-folder basis. Folder A 392 in web store 372 is set not to store any data. One  
4 reason for not storing the data for a folder is that the folder may be of minor significance for  
5 the users of a particular server. Because the folder is seldom if ever accessed, it is better to  
6 conserve the storage space that would otherwise be required for storing the folder's contents.

7       Nevertheless, by including the hierarchy on each server, it is possible to access the  
8 contents of each folder on any server. A request for the contents of Folder A 392 made to  
9 web store 372 will result in a redirect response that indicates where the contents of Folder  
10 A 392 may be accessed. Most likely, the access will be slower than would be provided by  
11 access to web store 372 because lower bandwidth communication links may be used, but  
12 this added burden is a minor imposition given the relative infrequency of access to Folder  
13 A 392 at back-end server 370. A folder within the storage hierarchy that does not include  
14 the folder's contents is said to be "ghosted."

15       When client system 310 requests App A 382 from front-end server 340, global  
16 catalog server 330 identifies a list of servers that includes back-end server 350 and back-end  
17 server 370. Front-end server 340 uses an authentication token associated with authentication  
18 credentials received from client system 310 as a key for performing a hashing operation over  
19 the list of servers. The hashing operation insures that requests for content are evenly  
20 distributed over the servers where the content is available and that a given request and  
21 authentication token will identify the same server each time the request is made (as long as  
22 the available servers do not change). By requesting data from the same back-end server  
23 each time a given request is made, any user-specific state information may be stored and  
24 recalled at the back-end server, such as a read/unread state for messages in a public folder.

1 Authentication tokens and authentication credentials may be stored in global catalog server  
2 330 and/or cached in front-end server 340.

3 Client system 310 requesting App B 384 from front-end server 340, identifies a  
4 single server, back-end server 360, from global catalog server 330. Like the mailbox  
5 examples from Figure 2, no hashing operation occurs because the content only may be  
6 requested from one back-end server. Thus, for both public and private content, it is possible  
7 that the hashing operation will not be necessary to identify a particular back-end server  
8 where the content request is directed.

9 Global catalog server 330 identifies back-end server 350, back-end server 360, and  
10 back-end server 370, in response to a request for the contents of Folder A 392 from  
11 front-end server 340. If the hashing operation identifies back-end server 350 or back-end  
12 server 360, the request is directed to the identified back-end server as described above. But,  
13 if back-end server 370 is identified, front-end server 340 must perform additional work in  
14 order to direct the request to a back-end server that can satisfy the request because the  
15 requested content is ghosted. When front-end server 340 directs the request to back-end  
16 server 370, back-end server 370 generates a redirect response. The contents of the redirect  
17 response depend on whether or not front-end 340 has modified the request to indicate that  
18 the response is from a front-end server. If so, the redirect response includes a list of  
19 back-end servers that can satisfy the request for the ghosted folder. Otherwise, the redirect  
20 response includes a single back-end server capable of satisfying the request. U.S.  
21 Application No. \_\_/\_\_\_\_\_, filed on the same day herewith, and entitled "Selecting a Server  
22 to Service Client Requests," which is hereby incorporated by reference, provides additional  
23 information on identifying a back-end server when public content is ghosted on a back-end  
24 server.

1        Figures 4A and 4B are flowcharts illustrating exemplary methods for redirecting  
2 requests for content stored at a back-end server according to the present invention. The  
3 dashed line between blocks 442 and 452 of Figure 4A indicates that the acts and steps above  
4 the line are performed at a front-end server whereas the acts and steps below the line are  
5 performed at a back-end server. A similar dashed line is included in Figure 4B. Returning  
6 to Figure 4A, block 412 represents the act of the front-end server receiving a request for  
7 content. As part of a step for authenticating the client (420), the front-end server may  
8 perform the acts of requesting client authentication credentials (422) and receiving the  
9 proper client authentication credentials (424) back from the client.

10        The front-end server adds a front-end indicator to the request (432) so that a  
11 back-end server receiving the request will know that it was sent by a front-end server rather  
12 than by a client. The step for querying (440A and 440B of Figure 4B) a back-end server for  
13 the content may include the acts of directing the request to a particular back-end server (442)  
14 and receiving a redirect response (444 of Figure 4B) from the particular back-end server. In  
15 generating the redirect response, the back-end server performs the following acts. The  
16 back-end server receives the request for content (452) and, in decision block 454, examines  
17 the request for a front-end indicator. If no front-end indicator is present, the redirect  
18 response includes a single redirect back-end server (462). With a front-end indicator  
19 present, decision block 456 determines if multiple back-end servers can satisfy the request  
20 for content. Here too, if only one back-end server is identified, the redirect response  
21 includes a single redirect back-end server (462). However, if multiple back-end servers are  
22 able to satisfy the content request, the redirect response includes a list of redirect back-end  
23 servers (458). After the one or more redirect back-end servers have been identified, the  
24 redirect response is sent to the front-end server (464 of Figure 4B).



000001 91262960

1 As described earlier, as part of a step for querying the back-end server for requested  
2 content (440A of Figure 4A and 440B), the front-end performs the act of receiving the  
3 redirect response from the back-end server. In decision block 466, the front-end server  
4 determines if the redirect response includes a list of back-end servers or a particular  
5 back-end server. If a list was received, the front-end server receives an authentication token  
6 (472) and hashes the list (474) with the authentication token as a key to identify a particular  
7 back-end server. The acts of receiving an authentication token (472) and hashing a list of  
8 back-end servers (474) may be included within a step for distributing a request for content to  
9 one of the back-end servers in the list. The authentication token may be identified during  
10 the step for authenticating the client (420 of Figure 4A).

11 Once a particular back-end server is identified, the front-end server redirects the  
12 request to the particular back-end server (482) and receives the requested content from the  
13 redirect back-end server (484). The acts of redirecting the request (482) and receiving the  
14 requested content (484) are acts that may occur in a step for retrieving the requested content  
15 from a particular redirect back-end server. After the content is received, the front-end server  
16 sends the requested content to the client system (492).

17 The present invention may be embodied in other specific forms without departing  
18 from its spirit or essential characteristics. The described embodiments are to be considered  
19 in all respects only as illustrative and not restrictive. The scope of the invention is,  
20 therefore, indicated by the appended claims rather than by the foregoing description. All  
21 changes which come within the meaning and range of equivalency of the claims are to be  
22 embraced within their scope.

23 What is claimed and desired to be secured by United States Letters Patent is:  
24

1 1. In a computerized system that includes a client system, a front-end server, and  
2 one or more back-end servers, all interconnected with a communication link, wherein the  
3 client system periodically accesses content stored on the one or more back-end servers  
4 through the front-end server, and wherein over time the content may be moved from one  
5 back-end server to another or may appear to be stored at a back-end server when in fact the  
6 content is not stored at that back-end server, a method of transparently redirecting a request  
7 for the content such that the client system is unaware of the redirection, the method  
8 comprising the front-end server performing the acts of:

- 9 receiving a request for the content from the client system;
- 10 directing the request to a particular back-end server;
- 11 receiving from the particular back-end server, a redirect response identifying
- 12 one or more other back-end servers where the content is stored;
- 13 automatically and without client system intervention, redirecting the request
- 14 to a redirect back-end server, the redirect back-end server being one of the one or
- 15 more other back-end servers identified in the redirect response;
- 16 receiving the requested content from the redirect back-end server; and
- 17 sending the requested content to the client system.

18  
19 2. The method as recited in claim 1 further comprising the act of adding a front-end  
20 indicator to the request in order to indicate that the front-end server is making the request on  
21 behalf of the client system.

22  
23 3. A method as recited in claim 2 wherein the front-end indicator is added to a  
24 hypertext transfer protocol User Agent header.

1  
2 4. A method as recited in claim 2 wherein the redirect response identifies a list of  
3 back-end servers where the content is stored.

4  
5 5. A method as recited in claim 4 wherein the list of back-end servers is identified  
6 in a hypertext transfer protocol 305 Use Proxy response from the particular back-end server.

7  
8 6. A method as recited in claim 4 further comprising the acts of:  
9 requesting authentication credentials from the client system; and  
10 receiving proper authentication credentials from the client system.

11  
12 7. A method as recited in claim 6 further comprising the acts of:  
13 receiving an authentication token that is associated with the authentication  
14 credentials; and  
15 using the authentication token as a key for a hash operation to identify the  
16 redirect back-end server from the list of back-end servers identified in the redirect  
17 response.

18  
19 8. A method as recited in claim 1 wherein the redirect response identifies a single  
20 back-end server where the content is stored.

21  
22 9. A method as recited in claim 8 wherein the single back-end server is identified in  
23 either a hypertext transfer protocol 301 Moved Permanently or 302 Moved Temporarily  
24 response from the particular server.

004007.00 97/05/2000

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24

10. A method as recited in claim 1, further comprising the acts of:

receiving the requested content from the redirect back-end server; and

sending the requested content to the client system.

1 11. In a computerized system that includes a client system, a front-end server, and  
2 one or more back-end servers, all interconnected with a communication link, wherein the  
3 client system periodically accesses content stored on one or more back-end servers through  
4 the front-end server, and wherein over time the content may be moved from one back-end  
5 server to another or may appear to be stored at a back-end server when in fact the content is  
6 not stored at that back-end server, a method of redirecting a request for the content directed  
7 to a particular back-end server when the content is not stored at the particular back-end  
8 server, the method comprising the back-end server performing the acts of:

9 receiving a content request from the client system through the front-end  
10 server, the content request including a front-end indicator in order to indicate that the  
11 front-end server is making the content request on behalf of the client system;

12 examining the content request for the front-end indicator;

13 the front-end indicator having been present in the content request, creating a  
14 redirect response to the content request that includes a list of one or more redirect  
15 back-end servers where the content is stored; and

16 sending the redirect response to the front-end server so that the front-end  
17 server can redirect the request to the one or more redirect back-end servers.

18  
19 12. A method as recited in claim 11 wherein the front-end indicator is added to a  
20 hypertext transfer protocol User Agent header.

21  
22 13. A method as recited in claim 11 wherein the list of one or more redirect back-end  
23 servers is identified in a hypertext transfer protocol 305 Use Proxy response from the  
24 particular back-end server.

004007 " 97 26 29 60

14. In a computerized system that includes a client system, a front-end server, and one or more back-end servers, all interconnected with a communication link, wherein the client system periodically accesses content stored on the one or more back-end servers through the front-end server, and wherein over time the content may be moved from one back-end server to another or may appear to be stored at a back-end server when in fact the content is not stored at that back-end server, a method of transparently redirecting a request for the content such that the client system is unaware of the redirection, the method comprising the front-end server performing:

- an act of receiving a request for the content from the client system;
- a step for querying a particular back-end server for the requested content, wherein the response to the query identifies one or more other back-end servers where the content is stored;
- a step for, automatically and without user intervention, retrieving the requested content from a redirect back-end server, the redirect back-end server being one of the one or more other back-end servers identified in the redirect response; and
- an act of sending the requested content to the client system.

15. A method as recited in claim 14 further comprising a step for authenticating the client system.

16. A method as recited in claim 15 wherein the query response identifies a list of back-end servers where the content is stored, the method further comprising a step for distributing the request to the redirect back-end server based on the client system authentication.

17. A method as recited in claim 14 wherein the query response identifies a single back-end servers where the content is stored.

1 18. In a computerized system that includes a client system, a front-end server, and  
2 one or more back-end servers, all interconnected with a communication link, wherein the  
3 client system periodically accesses content stored on the one or more back-end servers  
4 through the front-end server, and wherein over time the content may be moved from one  
5 back-end server to another or may appear to be stored at a back-end server when in fact the  
6 content is not stored at that back-end server, a method of transparently redirecting a request  
7 for the content such that the client system is unaware of the redirection, comprising the acts  
8 of:

9 the front-end server receiving a request for the content from the client  
10 system;

11 the front-end server directing the request to a particular back-end server;

12 the particular back-end server receiving the request from the front-end server;

13 the particular back-end server creating a redirect response that identifies one  
14 or more other back-end servers where the content is stored; and

15 the front-end server automatically and without client system intervention,  
16 redirecting the request to a redirect back-end server, the redirect back-end server  
17 being one of the one or more other back-end servers identified in the redirect  
18 response.

19  
20 19. A method as recited in claim 18 further comprising the act of the front-end server  
21 adding a front-end indicator to the request in order to indicate to the particular back-end  
22 server that the front-end server is making the request on the behalf of the client system.  
23  
24



004007 " 9 T 2 6 2 5 6 0

1 20. A method as recited in claim 19 wherein the redirect response from the particular  
2 back-end server identifies a list of back-end servers where the content is stored.

3  
4 21. A method as recited in claim 20 further comprising the acts of:

5 the front-end server requesting authentication credentials from the client  
6 system; and

7 the front-end server receiving proper authentication credentials from the  
8 client system.

9  
10 22. A method as recited in claim 21 further comprising the acts of:

11 the front-end server receiving an authentication token that is associated with  
12 the authentication credentials; and

13 the front-end server using the authentication token as a key for a hash  
14 operation to identify the redirect back-end server from the list of back-end servers  
15 identified in the redirect response.

16  
17 23. A method as recited in claim 18 wherein the redirect response from the particular  
18 back-end server identifies a single back-end server where the content is stored.

19  
20 24. A method as recited in claim 18, further comprising the acts of:

21 the front-end server receiving the requested content from the redirect  
22 back-end server; and

23 the front-end server sending the requested content to the client  
24 system.

25. In a computerized system that includes a client system, a front-end server, and one or more back-end servers, all interconnected with a communication link, wherein the client system periodically accesses content stored on the one or more back-end servers through the front-end server, and wherein over time the content may be moved from one back-end server to another or may appear to be stored at a back-end server when in fact the content is not stored at that back-end server, a computer program product for implementing a method of transparently redirecting a request for the content such that the client system is unaware of the redirection, comprising:

a computer readable medium for carrying machine-executable instructions for implementing the method; and

wherein said method is comprised of machine-executable instructions for the front-end server performing the acts of:

- receiving a request for the content from the client system;
- directing the request to a particular back-end server;
- receiving from the particular back-end server, a redirect response identifying one or more other back-end servers where the content is stored;
- automatically and without client system intervention, redirecting the request to a redirect back-end server, the redirect back-end server being one of the one or more other back-end servers identified in the redirect response;
- receiving the requested content from the redirect back-end server; and
- sending the requested content to the client system.

30. A computer program product as recited in claim 25, wherein the redirect response identifies a single back-end server where the content is stored.

1 31. In a computerized system that includes a client system, a front-end server, and  
2 one or more back-end servers, all interconnected with a communication link, wherein the  
3 client system periodically accesses content stored on one or more back-end servers through  
4 the front-end server, and wherein over time the content may be moved from one back-end  
5 server to another or may appear to be stored at a back-end server when in fact the content is  
6 not stored at that back-end server, a computer program product for implementing a method  
7 of redirecting a request for the content directed to a particular back-end server when the  
8 content is not stored at the particular back-end server, comprising:

9 a computer readable medium for carrying machine-executable instructions  
10 for implementing the method; and

11 wherein said method is comprised of machine-executable instructions for the  
12 particular back-end server performing the acts of:

13 receiving a request for the content from the client system through the  
14 front-end server, the request including a front-end indicator in order to  
15 indicate that the front-end server is making the request on behalf of the client  
16 system;

17 examining the content request for the front-end indicator;

18 the front-end indicator having been present in the content request,  
19 creating a redirect response to the request that includes a list of one or more  
20 redirect back-end servers where the content is stored; and

21 sending the redirect response to the front-end server so that the  
22 front-end server can redirect the request to the one or more redirect back-end  
23 servers.  
24

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24

32. A method as recited in claim 31 wherein the front-end indicator is added to a  
hypertext transfer protocol User Agent header.

33. A method as recited in claim 31 wherein the list of one or more redirect back-end  
servers is identified in a hypertext transfer protocol 305 Use Proxy response from the  
particular back-end server.

**ABSTRACT OF THE DISCLOSURE**

Methods and computer program products for transparently redirecting a request for content such that a client system is unaware of the redirection. A client requests content through a front-end server that provides a single point of access for content stored on one or more back-end servers. The single point of access makes it so the client does not know and does not care which particular back-end server stores the requested content. When a back-end server provides a redirect response for content that the back-end server does not store, the front-end server receives the redirect response and reissues the request to a server identified in the redirect response. A front-end server indicator is added to requests so that the back-end server knows the request is from a front-end server. This allows the back-end server to provide a list of servers in the redirect response without causing errors in clients making direct requests to back-end servers when the clients are unable to process a list of servers contained in a redirect response.

G:\DATA\PAT\WORDPAT\13768.169.doc

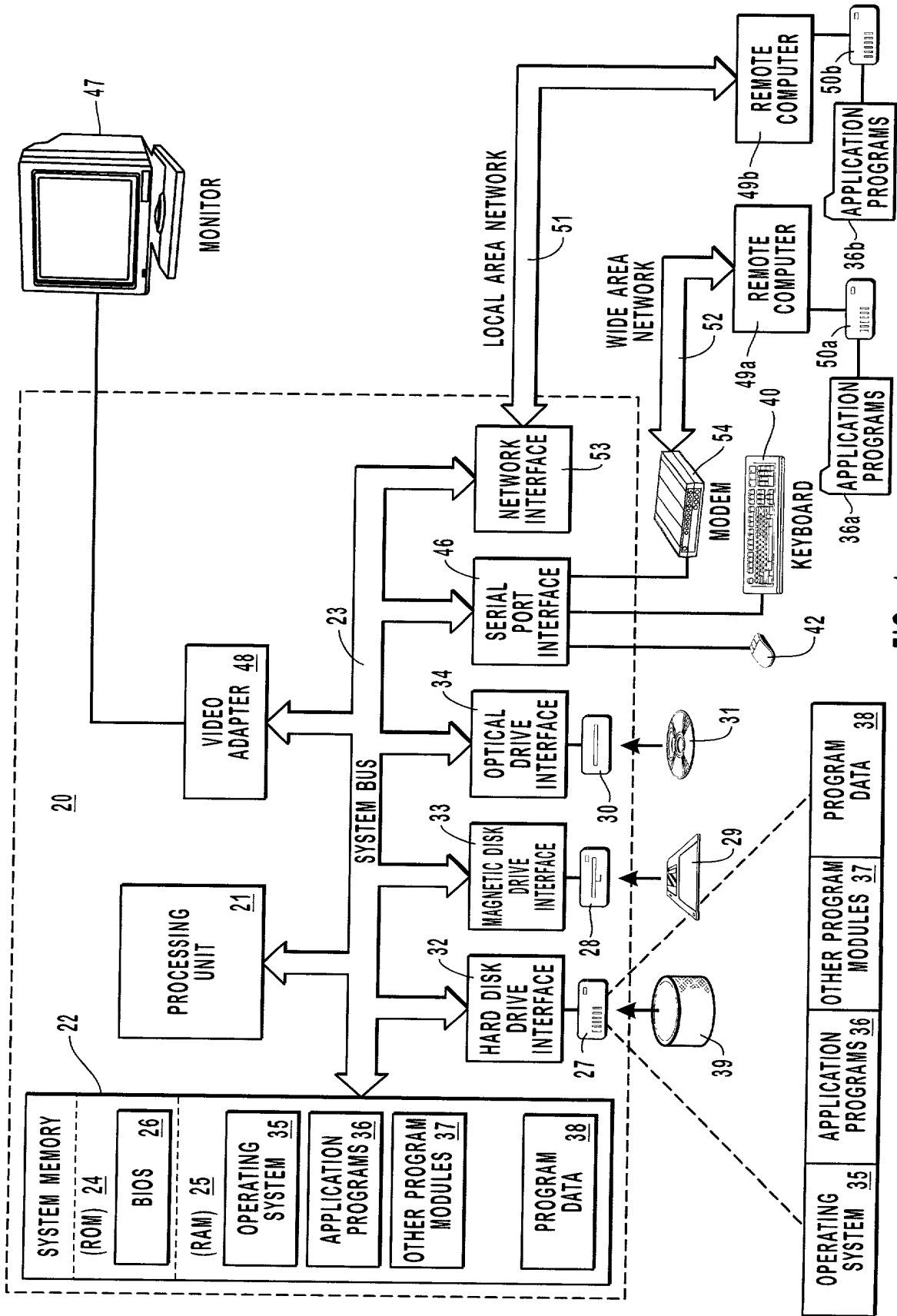


FIG. 1

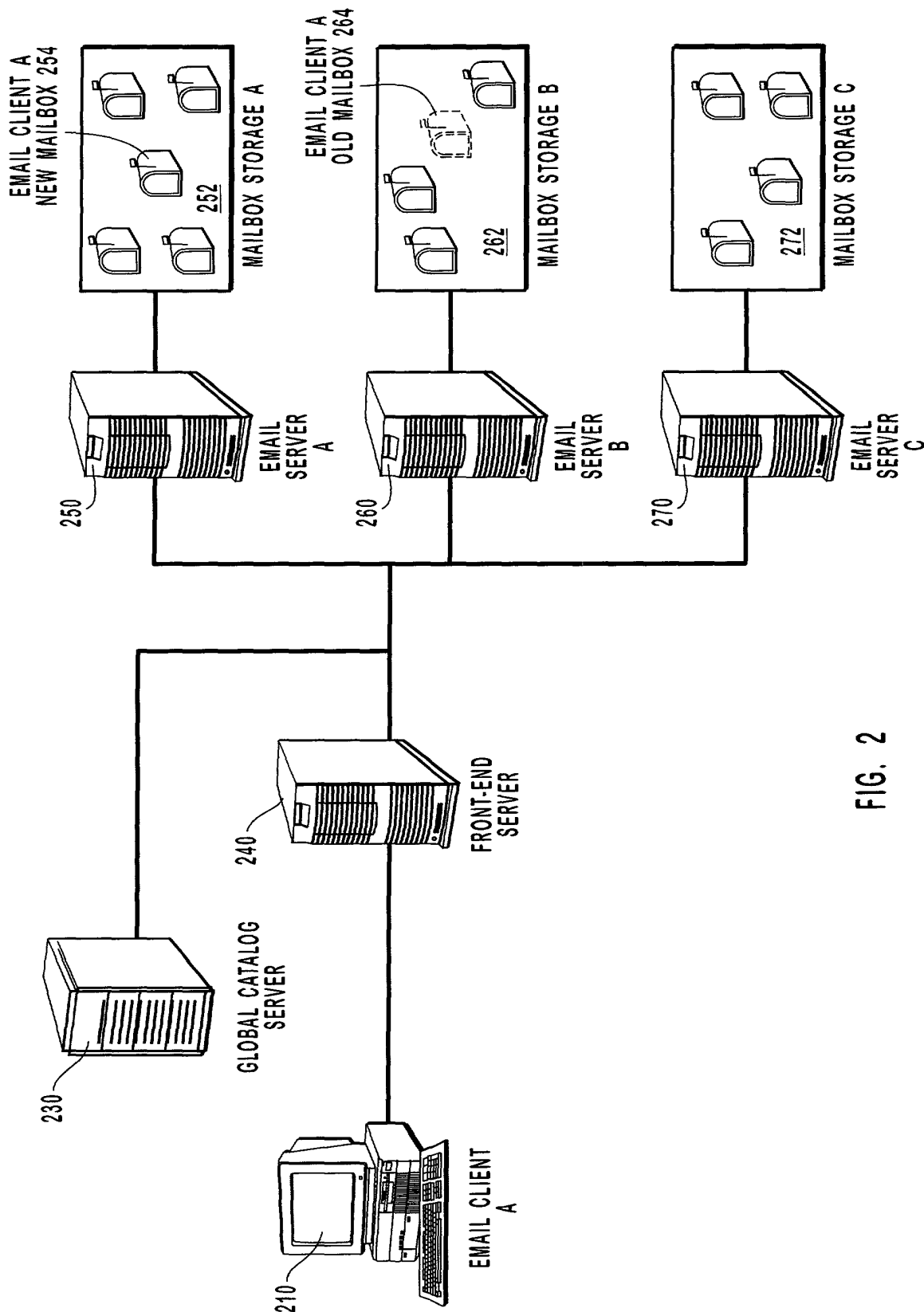


FIG. 2



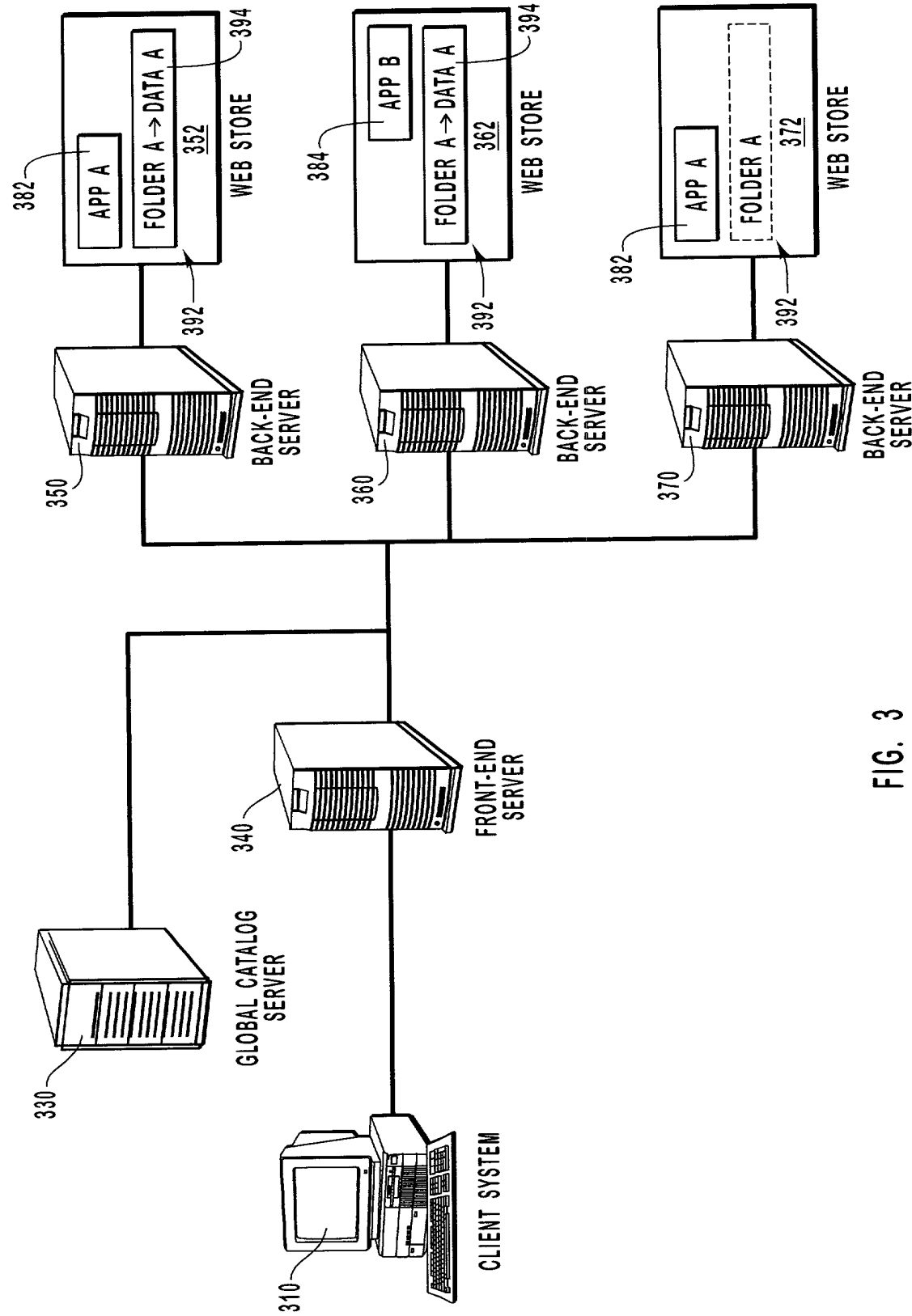


FIG. 3

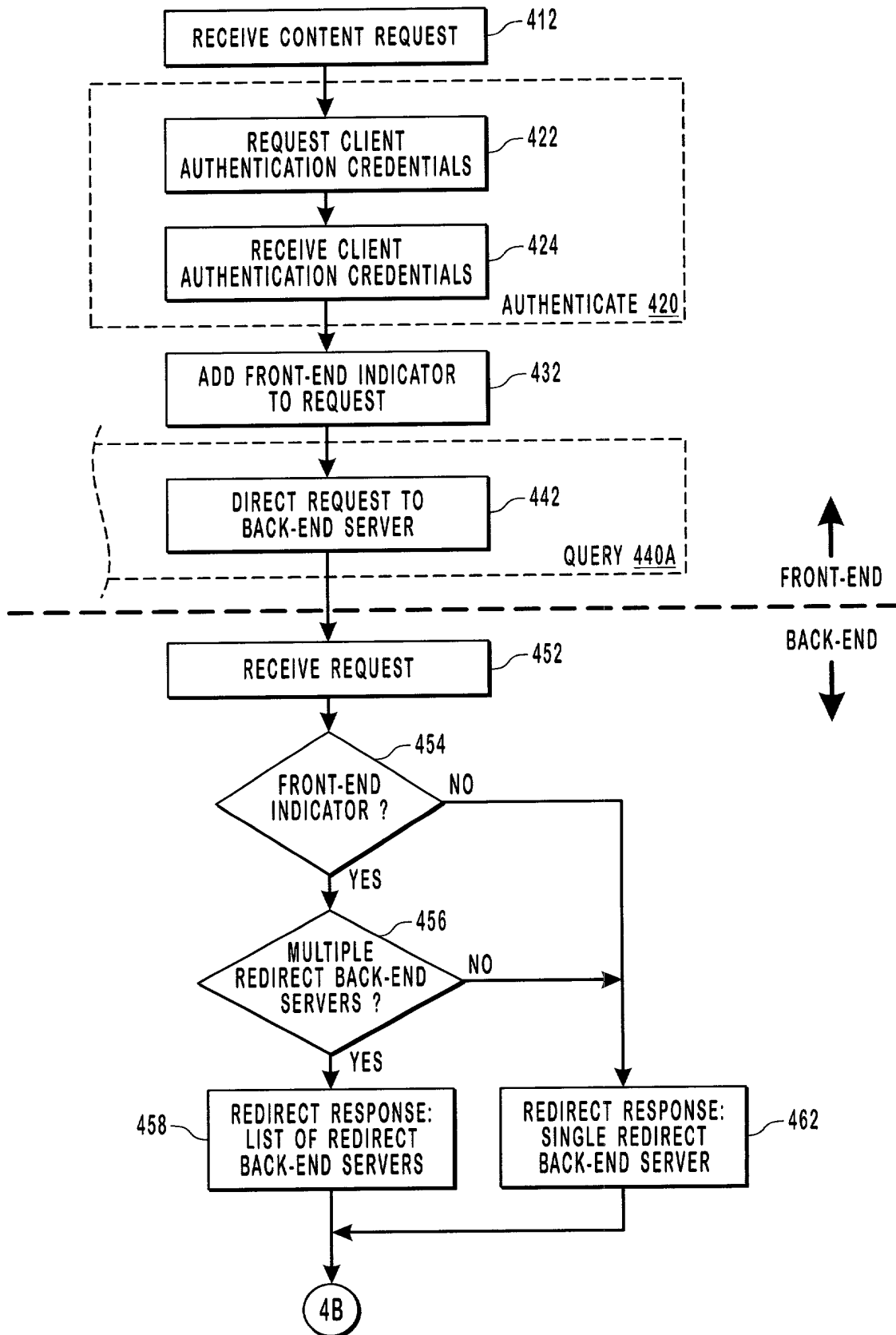


FIG. 4A

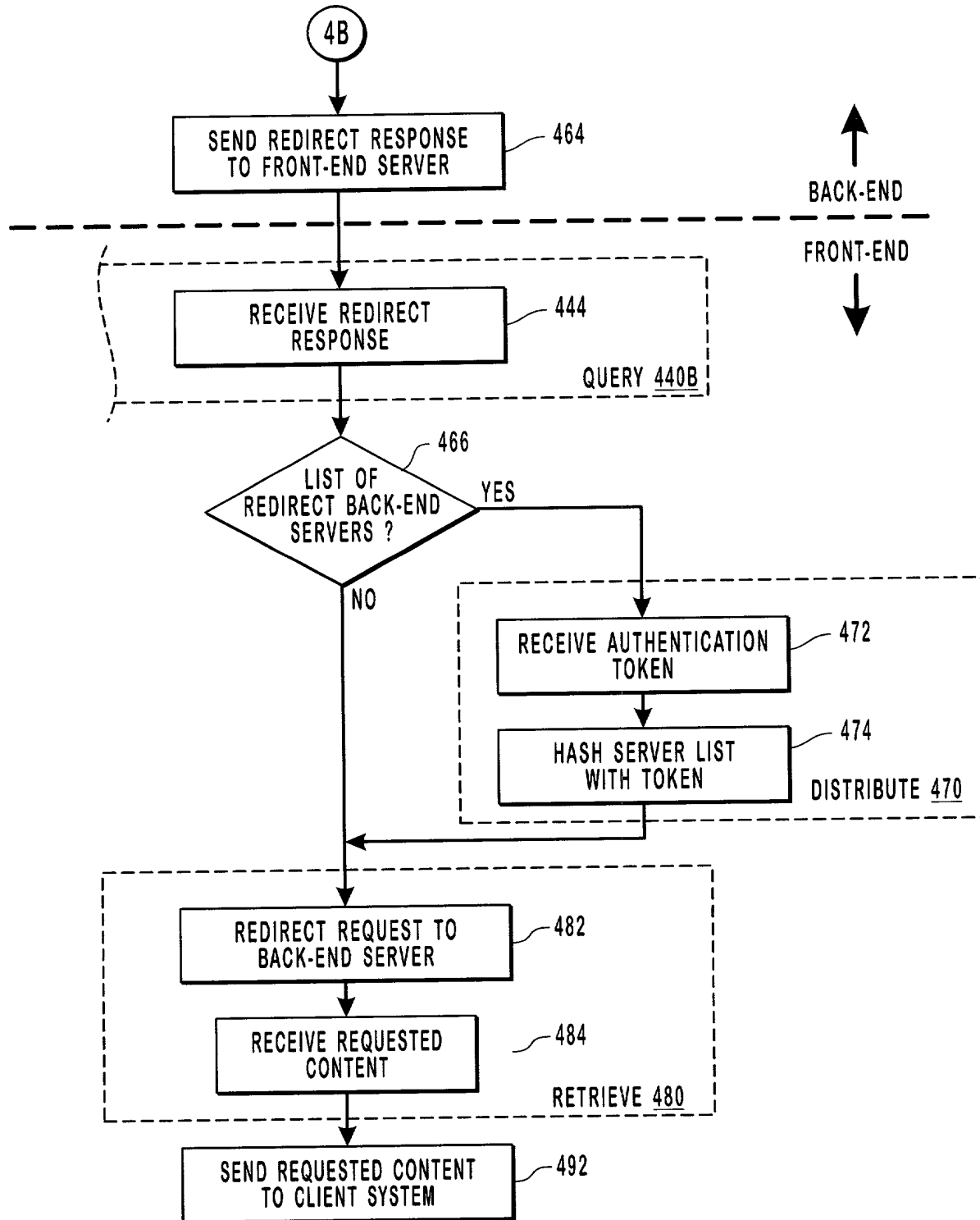


FIG. 4B

**ATTACHMENT**

All correspondence and telephonic communications relating to this patent application  
should be directed to:

RICK D. NYDEGGER  
WORKMAN, NYDEGGER & SEELEY  
1000 Eagle Gate Tower  
60 East South Temple  
Salt Lake City, Utah 84111  
Telephone: (801) 533-9800  
Facsimile: (801) 328-1707

004007 " 9746960